Systems Development in Information Systems Research

Jay F. Nunamaker, Jr.
Department of Management Information Systems
University of Arizona
Tucson, AZ 85721
602-621-2748
BITNET: NUNAMAKER@ARIZMIS

Minder Chen
Department of Decision Sciences
George Mason University
Fairfax, VA 22030
703-323-2758
BITNET: MCHEN@GMUVAX

Abstract

In this paper, the authors critically review the systems development in information systems (IS) research. Several classification schemes of research are described and systems development is identified as a developmental, engineering, and formulative type of research. A framework of research is proposed to explain the dual nature of systems development as a research methodology and a research domain in IS research. Progress in several disciplinary areas is reviewed to provide a basis to argue that systems development is a valid research methodology. A systems development research process is presented from a methodological perspective. Software engineering, the basic method in applying the systems development research methodology, is then discussed. A framework to classify IS research domain and various research methodologies in studying systems development is presented. It is the authors' belief that systems development and empirical research methodologies are complementary to each other. An integrated multi-dimension and multi-methodology approach will generate fruitful research results in IS research.

1 Research and Its Classification

There is a lot of confusion about IS as an academic discipline. "What constitutes valid IS research in terms of its research domain and research methodology?" is the most confusing and debatable question. Research is a "systematic, intensive study directed toward fuller scientific knowledge of the subject studied" [Blake, 1978, p. 3]. By this definition, IS research is no different from other types of research. However, by classifying research according to its domains and purposes, as well as the processes and tools used will help us to understand where IS research stands. Although these classification schemes overlap to some extent, they differ in their focus. The following are major research classification schemes found in the literature:

1. **Basic and applied research.** Basic research in developing and testing theories and hypotheses is undertaken in response to the intellectual interests of the researcher, rather than for practical reasons. Applied research is the application of knowledge to solve problems of immediate concern [Blake, 1978; Bailey, 1982].

2. **Scientific and engineering research.** There is no logical distinction between the methods used by the engineer and the pure scientist. Both are concerned with confirming their theoretical predictions. However, they do differ in the scale of their experiments and their motives. In the engineering approach, the artistry of design and the spirit of "making something work" are also essential [Davies, 1973].

3. **Evaluative and developmental research.** There are two types of research that are directed to solving problems: evaluative and developmental [Ackoff, Gupta, and Minas, 1962]. The developmental type of research "involves the search for (and perhaps construction or synthesis of) instructions" which yield a better course of action [Ackoff, Gupta, and Minas, 1962, p. 24]. Developmental research has largely been ignored by researchers of the social and behavioral sciences. However, without research efforts directed toward developing new solutions, there will be very little opportunity left for evaluative research.

4. **Research and development.** Development is the systematic use of scientific knowledge directed toward the production of useful materials, devices, systems, or methods, including design and development of prototypes and processes [Blake, 1978]. Hitch and McKean [1960] classified development work as: exploratory, advanced, engineering, and operational development. The first three types of development can also be labeled as applied research. The authors believe that **"without development, research has no use; without research, development has no base."**

5. **Formulative and verificational research.** The goal of formulative research (also called exploratory research) is to identify problems for more precise investigation, to develop hypotheses, as well as to gain insights and to increase

0073-1129/90/0000/0631$01.00 © 1990 IEEE

familiarity with the problem area. The goal of verification research is to obtain evidence to support and refute formulated hypotheses [Grosof and Sardy, 1985].

Systems development as a research methodology falls into the category of applied science and belongs to the engineering, developmental, and formulative type of research. A framework of research is described in Section 2 to distinguish the research methodology from the research domain. The importance of systems development research methodology to the enrichment of human knowledge is addressed in Section 3. However, **academic research** generally focuses on **the extension of human knowledge and demonstration of technical excellence** [Blake, 1978]. The development of systems, especially the development of information systems, has to follow a certain research process and conform to some criteria to be qualified as academic research. A process which constitutes the systems development research methodology is discussed in Section 4. Just as statistics provide a method for conducting empirical research, software engineering is the primary method for conducting (software) systems development research. The discussion of software engineering can be found in Section 5. In Section 6, the authors identify several dimensions to classify systems development as a research domain and demonstrate how various methodologies can be used in systems development research.

2 Research Domain and Research Methodology: A Framework

Systems development, especially the development of software systems, is a research domain as well as a research methodology. Questions, such as "Does the development of a software system constitute a research project (in the academic sense)?", have been frequently raised. The dual nature of systems development usually is the cause of confusion and is discussed in this section. Definitions of research domain and research methodology are provided for clarification.

A **research domain** is the subject matter under study in a research project. A **research methodology** consists of the combination of the process, methods, and tools which are used in conducting research in a research domain. In Figure 1, a framework of research is proposed to explain the relationship between research domains and research methodologies. The body of knowledge includes both research domains and research methodologies. A research process involves understanding of research domains, finding out meaningful research questions, and applying valid research methodologies to address these questions. Results from a good research project can contribute to the body of knowledge both by expanding knowledge in a given domain and by enlarging applicable methodologies in the domain.

For valid research, the authors believe that the research method is no more important than the research question. Research methods are means of finding truth in research domains. Without an understanding of a research domain, researchers might ask a wrong question or formulate a meaningless hypothesis. No matter what research methods they apply, incorrect or irrelevant questions can only lead researchers to inappropriate conclusions. Benbasat [1984] identifies case study, field study, field experiment, laboratory experiment, and sample survey as empirical research strategies for management support systems [1984]. In Galliers and Land's [1987] taxonomy for IS research methodologies, some newer approaches such as action research have been included. However, neither of them includes systems development as one of the IS research methodologies. Galliers and Land challenged the use of traditional approaches (empirical methodologies) for IS research by stating that they "may well be academically acceptable and internally consistent, all too often they lead to inconclusive and inapplicable results." Systems development as a research methodology can be used not only as a means of better understanding a research domain, but can sometimes even change the processes and products in a research domain.

3 Systems Development as a Research Methodology

The systems development methodology is an age-old method and process that human beings use to study nature and to create new things. Table 1 lists a few examples of how systems development efforts have contributed to several research domains.

In the airplane design area, the Wright brothers built the first airplane before the aerodynamics field had been created. Aerodynamics and aerostatics are branches of engineering that were created by studying model airplanes built in the laboratory and learning from experiences gained through building real airplanes. The aircraft industry is now using the most advanced Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM) tools to design next generation airplanes. These CAD/CAM tools have encoded theories developed in aerodynamics and heuristics learned from building real systems. The use of CAD/CAM tools has saved the airplane industry millions of dollars by improving the performance of new airplanes. The pattern of this research progress is: 1) build a system, 2) develop theories and principles from observing behavior, 3) encode expertise in software tools for easy access, and 4) use these tools to help the development of new systems.

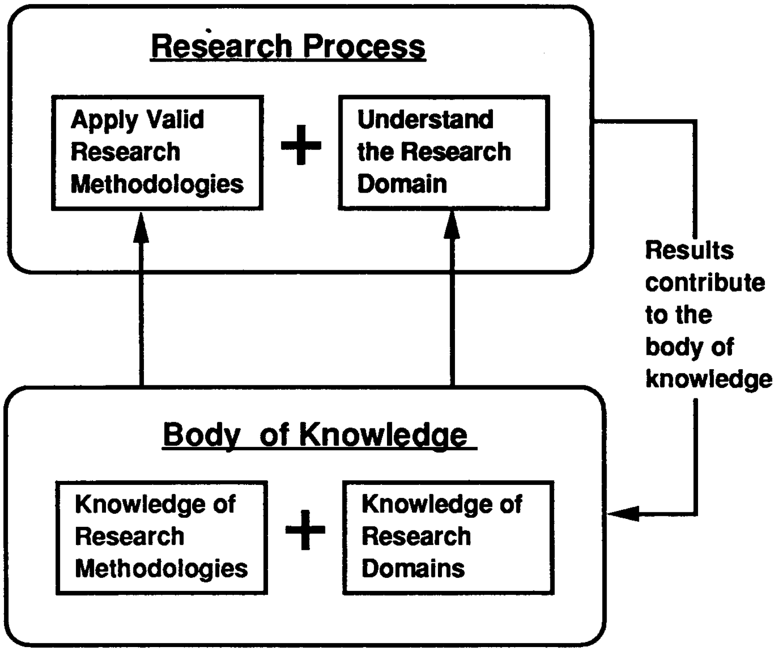

Research Process
Apply Valid Research Methodologies
+
Understand the Research Domain
Results contribute to the body of knowledge
Body of Knowledge
Knowledge of Research Methodologies
+
Knowledge of Research Domains

Figure 1: A Framework of Research

Table 1: Example Contributions of Systems Development to Different Research Areas

Research Domain	Progress
Airplane Design	- The Wright brothers designed the first airplane (1903) - Development of aerodynamics and aerostatics - CAD/CAM for airplane design and manufacturing (1980s)
Memory Management in Computer Systems	- Real memory management - Simulation of memory usage - Virtual memory management - Mathematical models of memory usage
Software Development Methods and Tools	- Structured programming (Early 1970s) - Structured design (Mid 1970s) - Structured analysis (Late 1970s) - Computer-aided software engineering (CASE) (Early 1980s) - Empirical studies of systems development (Early 1980s) - Empirical study of CASE technology (Late 1980s)
Computer-Supported Cooperative Works	- Electronic mail (Late 1960s) - Teleconferencing (Early 1970s) - Group decision support systems (GDSS) (Early 1980s) - Empirical evaluation of groupware (Mid 1980s) - Intelligent E-Mail (Mid 1980s) - Integrated work group systems (1990s)

In the case of memory management in computer systems, various memory management techniques were developed from previous systems building experiences and evaluation of the systems that were built [Deitel, 1984]. Simulation was first used to study the pattern of memory usage of various memory management schemes. Later, mathematical models were developed to study their performance. In one instance, Peter Denning developed the Working-Set Model of program behavior from the **observation** of the **locality** phenomena in a paging memory system which was developed by students at the Massachusetts Institute of Technology [Denning, 1968]. Locality is the phenomenon that programs tend to reference main memory in non-uniform and highly localized patterns. "**It is an empirical (observed) property rather than theoretical one**" [Deitel, 1984; p. 222]. A working set memory management policy was proposed to improve systems performance by preventing possible thrashing. The pattern of this research progress is: 1) build a system, 2) observe its behavior, 3) develop a mathematical model to explain the behavior of the system, and 4) formulate a new mechanism to improve the system performance.

In the area of software development methods and tools, structured programming, structured design, and structured analysis have been introduced in sequence by practical experiences learned from developing real systems [Martin and McClure, 1988]. Empirical studies of programming were motivated by the publication of Gerald Weinberg's *The Psychology of Programming* in 1971. Empirical studies and comparison of various system analysis and design methods began only at early 1980s [Hoffer, 1982]. The development of Computer-Aided Software Engineering (CASE) tools also became very active in early 1980s, however, empirical studies of this technology did not start until the late 1980s [Norman and Nunamaker, 1989; Orlikowski, 1988]. The patterns of progress in this area are: 1) learn from the development of software systems, 2) formulate structured methodologies to improve systems development process, 3) develop automated tools to support the use of structured methodologies, and 4) study the use of various design methods and tools empirically.

In the computer-supported cooperative work (CSCW) area [Greif, 1988], the advent of electronic mail, teleconferencing [Johansen and Bullen, 1984], and group decision support systems [Huber, 1984; Dennis et al., 1988] led to research studying the effects of these CSCW tools on organizational structures and dynamics, as well as individual and group behaviors which use them. Such empirical studies became possible because of these emerging technologies. The patterns of progress in CSCW are: 1) introduce systems (i.e., electronic mail and teleconferencing) to support collaborative work which increases the demand for more technology support in human collaboration, 2) develop more new systems such as GDSS, 3) study the use of these systems empirically, and 4) apply results from empirical studies to improve existing systems.

Some basic principles in systems development emerge from the examples discussed above and can be summarized as follows:

1. One should not assume that he already knows too much about a domain. Building a prototype system always helps to study and to understand a research domain. Researchers may learn more about all aspects of a domain from observing the prototype behavior as well as from building it. This approach will result in a successive refinement to the prototype system and its building process.

2. Newly developed systems may change the processes and concepts in a domain such that expand the horizon of human knowledge about their surroundings.

3. The use of tools (e.g., CASE and CAD/CAM) to support systems development has also been found very useful in amplifying human intelligence and in transferring knowledge for wider use.

4. Systems development research methodology can be used in conjunction with other research methodologies (e.g., laboratory experiments, field studies, and case studies). The empirical study of the development process and method or of the use of the prototype system may provide valuable feedback for more effectively designing a system.

4 Process of Systems Development Research Methodology

Methodology is the philosophy of the **research process** which "includes the **assumptions and values** that serve as a rationale for research and the **standards or criteria** the researcher uses for interpreting data and reaching conclusion" [Bailey, 1982, p. 26]. Research process, the heart of research methodology, is the application of scientific method to the complex task of discovering answers (solutions) to questions (problems) [Blalock and Blalock, 1982]. The research process in social and behavioral sciences can be summarized as follows [Bailey, 1982; Blalock and Blalock, 1982]: 1) choosing the research problem(s), 2) stating hypotheses, 3) formulating the research design, 4) gathering data, 5) analyzing data, and 6) interpreting the results so as to test hypotheses. The authors found a parallelism between the social (behavioral) and engineering (development) types of research, although the detailed methods and tools used may differ.

Figure 2 shows a research process of systems development from a methodological viewpoinf. Research issues which should be addressed in each stage are also identified. A systems building process (with emphasis on software development) consists of the following stages:

1. **Construct a conceptual framework.**
   Researchers should first justify the significance of research questions pursued. An ideal research problem is one that is new, creative, and important in the field. When the proposed solution of the research problem cannot be proven mathematically and tested empirically, or if it proposes a new way of doing things, researchers have to develop a system to demonstrate the validity of the solution, based on the suggested new methods, techniques, or design. Once the system has been built, researchers can study its performance and the phenomena related to its use to gain insights into the research problem. A clear definition of the research problem provides a focus for the research throughout the development process. The research question should be discussed in the context of an appropriate conceptual framework. Various disciplines should also be explored to find additional approaches and ideas which could be incorporated in the new system.

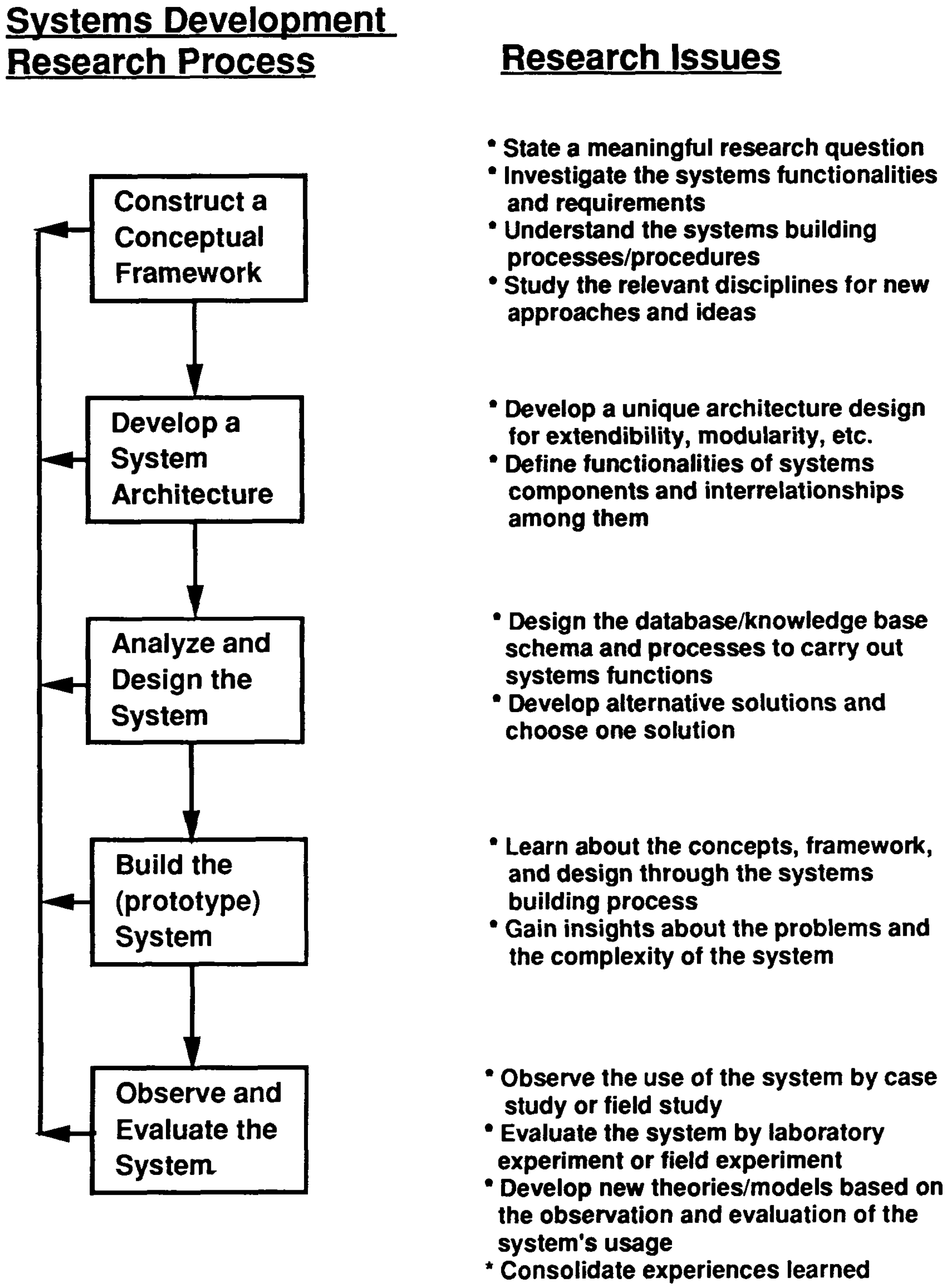
2. **Develop a system architecture.**
   A good system architecture provides a road map for the systems building process. It puts the system components into the correct perspective, specifies the system functionalities, and defines the structure relationships and dynamic interactions among system components. In the development type of research, researchers must identify the constraints given by the environment, state the objectives of the development efforts (i.e., the focus of the research), and define the functionalities of the resulting system to achieve the stated objectives. Requirements should be defined so that they are measurable and thus can be validated at the systems evaluation stage. In the empirical and evaluative type of research, formulating the research hypotheses is an important step in the research process. In the development type of research, researchers usually do not formulate an explicit hypothesis, but they do make assumptions about the research domain and the technical environment for developing the system. Researchers state the system requirements under the constraints of these assumptions and design and implement the system according to the requirements. Depending on the focus of the research, one might emphasize the new functionalities or innovative use interface features of the proposed new system rather than the throughput or the response time of the system.

3. **Analyze and design the system.**
   A research project's requirements may be driven by new functionalities envisioned by the researcher or may be determined partially by the research sponsor's requests. Design is the most important part of a system development process. Design involves the understanding of the studied domain, the application of relevant scientific and technical knowledge, the creation of various alternatives, and the synthesis and evaluation of proposed alternative solutions. Design specifications will be used as a blueprint for the implementation of the system. For a software development project, design of data structures, databases, or knowledge bases should be determined at this phase. The program modules and functions also should be specified at this time after alternatives have been proposed and explored and final design decisions been made.

4. **Build the system.**
   Implementation of a system is used to demonstrate the feasibility of the design and the usability of the functionalities of a system development research project. The process of implementing a working system can provide researchers insights into the advantages and disadvantages of the concepts, the frameworks, and the chosen design alternatives. The accumulated experiences and knowledge will be helpful in re-designing the system. Empirical studies of the functionalities and the usabilities can obe only performed after the system has been built.

5. **Observe and evaluate the system.**
   Once the system is built, researchers can test its performance and usability as stated in the requirement definition phase, as well as observe its impacts on individuals, groups, or organizations. The test results should be interpreted and evaluated based on the conceptual framework and the requirements of the system defined at the earlier stages. Development is an evolutionary process. Experiences gained from developing the system usually will lead to the further development of the system, or even the discovery of a new theory to explain observed new phenomena.

The use of system development as a research methodology in IS should conform to the following



Systems Development Research Process

Research Issues

Construct a Conceptual Framework

- State a meaningful research question
- Investigate the systems functionalities and requirements
- Understand the systems building processes/procedures
- Study the relevant disciplines for new approaches and ideas

Develop a System Architecture

- Develop a unique architecture design for extendibility, modularity, etc.
- Define functionalities of systems components and interrelationships among them

Analyze and Design the System

- Design the database/knowledge base schema and processes to carry out systems functions
- Develop alternative solutions and choose one solution

Build the (prototype) System

- Learn about the concepts, framework, and design through the systems building process
- Gain insights about the problems and the complexity of the system

Observe and Evaluate the System

- Observe the use of the system by case study or field study
- Evaluate the system by laboratory experiment or field experiment
- Develop new theories/models based on the observation and evaluation of the system's usage

* Consolidate experiences learned

Figure 2: A Research Process of Systems Development Research Methodology

criteria: 1) The purpose is to study an important phenomenon in areas of information systems through system building, 2) The results have significant contributions to the domain, 3) The system is testable against all the stated objectives and requirements, 4) The new system can provide better solutions to certain information system problems than existing systems, and 5) Experiences and design expertise learned from building the system can be generalized so that they can be used in other situations.

In every phase of the system development process, researchers gain insights about a domain that will lead to changing some design decisions made in previous phases. When the developed system is a software system, software engineering methods and techniques should be used to improve the quality of both the development process and the research results. This leads us to a discussion of software engineering as a method in systems development research.

5 Software Engineering: A Method for Systems Development Research

It is essential to understand what software is and the importance of software productivity in order to appreciate research efforts in software engineering [Nunamaker and Chen, 1987; Boehm, 1987]. New software systems developed in the information systems area definitely change the way people think about information systems and the way they solve information systems problems [Lyytinen, 1985]. For example, the advent of spreadsheet software and financial modeling languages makes decision support systems a feasible solution to managerial decision-making problems. The new hypertext systems will probably change the way people read and write as well as the way they think and communicate [Conklin, 1987]. Information systems is an applied discipline. If research in information systems fails to be applicable to the real world, then the research efforts are in vain [Galliers and Land, 1987].

Software, which is a critical part of modern information systems, can be broadly defined as [Freeman, 1987]: 1) the embodiment of the functions of a system, 2) the captured knowledge of an application area, and 3) the information produced during the system development process. Due to the complexity of a software system, its success relies on the application of rigid discipline in its development process, i.e., software engineering.

There is no generally agreed definition of software engineering, but the following definitions will serve as a basis for discussion:

1. Naur's definition: "The phrase **software engineering** was deliberately chosen as being provocative, in implying the need for software manufacture to be used on the types of theoretical foundations and practical disciplines that are traditionally in the established branches of engineering" [Naur, Randell, and Buxton, 1976, p. 9].

2. Vick's definition: In the preface of *Software Engineering Handbook*, Vick and Ramamoorthy [1984, p. IX] state that software engineering is used to "interpret and apply sound **engineering discipline and practice** to the design, development, testing, and maintenance of **software systems.**" It is not just "a collection of tools and techniques, it is **engineering** ... software engineering can learn from other engineering disciplines...."

3. Wegner's definition: Wegner [1983, p. 167] emphasizes the conceptual level constructs of software development, saying that "the paradigms of **software engineering** are those of conventional engineering modified to take into account the fact that **software is a conceptual rather than a physical product.**"

4. Definition in *IEEE Standard Glossary of Software Engineering Terminology*: "Systematic approach to the development, operation, maintenance, and retirement of software" [IEEE, 1983, p. 32].

5. Macro and Buxton's definition [1987, p. 3]: "The establishment and use of sound engineering principles and good management practice, and the evolution of applicable tools and methods and their use as appropriate, in order to obtain -- within known and adequate resource provisions -- software that is of high quality in an explicitly defined sense. "

In summary, software engineering has the following characteristics: 1) It is an engineering discipline, 2) It studies the methods, techniques, tools, processes, and management of the development of software systems, and 3) It is a systematic approach.

The development of (software) systems to conduct research can be traced back to the research paradigm of engineering schools. Engineering research paradigm has heavily influenced systems development research methodology. The engineers generally agree that "progress is achieved primarily by posing **problems** and systematically following the **process** to construct systems that solved them" [Denning et al., 1988, p. 4]. The principles of engineering (e.g., Roadstrum [1967]) are the foundation of systems development methodology.

6 Systems Development as a' Research Domain

As a research domain, systems development is one of the major areas of IS research. IS development environment and process are two of nine major factors in the IS research model proposed by Ives, Hamilton, and Davis [1980]. Information systems as a research domain can be classified according to the following four dimensions: 1) The group-size dimension is determined by the scale of the subject under study (e.g., individual, team, project, and company), 2) The process dimension includes each phase in the systems lifecycle (e.g., planning, analysis, design, implementation, usage, and maintenance), 3) The technical dimension encompasses systems development methods, techniques, and tools (e.g., fourth generation languages, CASE tools, as well as structured analysis and design methods), 4) the behavioral dimension ranges from the study of human cognition to organizational cultures. This four-dimension framework may be used to help IS researchers to classify existing IS research and to identify future research opportunities.

A multi-methodological approach will be useful to study various aspects of systems development processes and environments. The authors adapt Scott Morton's [1984] categories of Management Support Systems research to show how various research methodologies can be used in studying IS as a research domain:

1. **Build a (prototype) system.** "Building a prototype system is an engineering concept" [Scott Morton, 1984]. It has been widely used in systems development research. Researchers in systems development often conduct their research by building a system.

2. **Construct a method.** Parnas's [1979] paper on using modularization in systems design basically proposes a concept of how to build a software system with improved flexibility and comprehensibility in shorter time by using modularization. Some software engineering principles such as information hiding and hierarchical decomposition are derived from the concept presented in his paper. Booch's [1986] article on "Object-Oriented Development" is also an example of how to construct a new software design method, but at a more specific level.

3. **Develop a theory.** Halstead [1977] has developed a theory, called *software science*, that calculates the operators and operands of a program to estimate some properties of that program. This is a classic example of building a theory for systems development.

4. **Formulate a concept (i.e., a framework).** Research in this category suggests "a framework that is found useful in organization ideas and suggesting actions" [Scott Morton, 1984]. Nunamaker's and Chen's [1987] work on proposing a framework to study software productivity and reusable software components is an example.

5. **Conduct an empirical laboratory test.** Basili's paper "Experimentation in Software Engineering" has provided a framework for conducting experiments in software engineering [Basili, Selby, and Hutchens, 1986]. *Tutorial: Human Factors in Software Development* edited by Bill Curtis [1985] presents a broad overview and many examples of experimental research on human factors in systems development. Ledgard [1987] also discusses some examples of inherent difficulties and the possibility of misleading in conducting empirical study of software engineering.

6. **Conduct a real-world test or a survey.** Survey studies used in systems development research often focus on the evaluation of different development methods used in a real-world setting. Mahmood's [1987] paper on comparing the software development life cycle and prototyping methods is an example of using a survey study in software development domain. Norman and Nunamaker [1989] used the survey method to study the CASE productivity perceptions of software engineering professionals.

7. **Describe a case.** Developing a system is learning by doing. Knowledge gained from the development process can be consolidated into a case study, which describes the rationale, process, and experiences learned from developing a system. Orlikowski [1988] conducted a case study of implementation CASE tools in an organization with an emphasis on their impact on the IS workplace. In systems development research, the researcher who conducts a case study is usually actively participating in the development of the system among other activities such as action research [Gibson 1975]. This usually is not the case in a social science study.

8. **Declare the "truth."** Something very close to the declaration of truth is found in Dijkstra's letter to the editor of *Communications of the ACM* in which he declared that "go to statement considered harmful" [Dijkstra, 1968].

7 Conclusion

Systems development research methodology has been one of the major methodologies used for research at the Department of Management Information Systems (MIS) at the University of Arizona. A dominant theme in the MIS Department states that design is the key to IS and the emphasis is on rigorous IS development. Earlier research projects, such as PLEXSYS, focused on the building of an integrated environment for systems development [Konsynski and Nunamaker, 1982]. The need to allow users, managers, and systems developers to interact with each other in a group setting to elicit IS requirements led to the development of electronic meeting systems [Dennis et al, 1988] at the University of Arizona. The facilities have been used not only for facilitating the systems development process, but also have been used for a wide range of group meetings (e.g., business planning and knowledge acquisition). Several empirical studies have been conducted to validate the effectiveness of the electronic meeting systems [Nunamaker, Applegate, and Konsynski, 1987; [Easton, Vogel, and Nunamaker, 1989]. Using systems development research methodology as the core research methodology in conjunction with other research methodologies in various reference disciplines has been very successful in the University of Arizona's MIS program.

Systems development is not only an important research domain in IS but it can be a very useful research methodology in conducting IS research. It is the authors' belief that systems development and empirical research methodologies are complementary to each other. An integrated multi-dimension and multi-methodology approach will generate fruitful research results in IS research.

8 References

Ackoff, R. L., Gupta, S. K., and Minas, J. S., *Scientific Method*, John Wiley & Sons, Inc., 1962.

Bailey, K. D., *Methods of Social Research*, The Free Press, 1982.

Basili, V. R., Selby, R. W., and Hutchens, D. H., "Experimentation in Software Engineering," *IEEE Transactions on Software Engineering*, Vol. SE-12, No. 7, July 1986, pp. 733-743.

Benbasat, I., "An Analysis of Research Methodologies," in W. F. McFarlan (ed.), *The Information Systems Research Challenge*, Harvard Business School Press, 1984, pp. 47-85.

Blake, S. P., *Managing for Responsive Research and Development*, W. H. Freeman and Company, 1978.

Blalock, A. B. and Blalock, H. M., Jr., *Introduction to Social Research*, Prentice-Hall, second edition, 1982.

Boehm, B. W., "Improving Software Productivity," *IEEE Computer*, September 1987, pp. 43-57.

Booch, G., "Object-Oriented Development," *IEEE Transactions on Software Engineering*, Vol. SE-12, No. 2, February 1986, pp. 211-221.

Conklin, J., "A Survey of Hypertext", *IEEE Computer*, September 1987, pp. 17-41.

Curtis, B., (ed.) *Tutorial: Human Factors in Software Development*, IEEE Computer Society, 1985.

Davies, J. T., *The Scientific Approach*, second edition, Academic Press, 1973.

Deitel, H. M., *An Introduction to Operating Systems*, revised first edition, Addison-Wesley Publishing Co., 1984.

Denning, P. J., "The Working Set Model for Program Behavior," *Communications of the ACM*, Vol. 11, No. 5, May 1968, pp. 323-333.

Denning, P. J. et al., *Draft Report of the ACM Task Force on the Core of Computer Science*, DRAFT 3, February 2, 1988.

Dennis, A. R., George, J. F., Jessup, L. M., Nunamaker, J. F., Jr., and Vogel, D. R., "Information Technology to Support Electronic Meetings," *MIS Quarterly*, Vol. 12, 1988, pp. 591-618.

Dijkstra, E., "Go To Statements Considered Harmful," *Communications of the ACM*, Vol. 11, No. 3, March 1968, pp. 147-148.

Easton, A. C., Vogel, D. R., and Nunamaker, J. F., Jr., "Stakeholder Identification and Assumption Surfacing in Small Group: An Experiment Study," *The 22nd International Conference on System Sciences*, Hawaii, Janurary 1989, pp. 344-352.

Freeman, P. *Software Perspectives: The System Is the Message*, Addison-Wesley Publishing Co., 1987.

Galliers, R. D. and Land, F. F., "Choosing Appropriate Information Systems Research Methodologies," *Communications of the ACM*, Vol. 30, No. 11, November 1987.

Gibson, C. F., "A Methodology for Implementation

Research," in R. L. Shultz and D. P. Slevin (eds.) *Implementing Operations Research/Management Sciences*, New York: American Elsevier, 1975, pp. 53-73.

Greif, I., *Computer-Supported Cooperative Work: A Book of Readings*, Morgan Kaufmann Publishers, Inc., 1988.

Grosof, M. S. and Sardy, H., *A Research Primer for the Social and Behavioral Sciences*, Academic Press, Inc., 1985.

Halstead, M. H., *Elements of Software Science*, Elsevier North-Holland, 1977.

Hitch, C. J. and McKean, R. N., *The Economics of Defense in the Nuclear Age*, Harvard University Press, 1960.

Hoffer, J. A., "An Empirical Investigation into Individual Differences in Database Models," *Proceedings of the Third International Conference on Information Systems*, 1982, pp. 153-167.

Huber, G. P., "Issues in the Design of Group Decision Support Systems," *MIS Quarterly*, September, 1984, pp. 195-204.

IEEE Standard Glossary of Software Engineering Terminology, The Institute of Electrical and Electronics Engineering, Inc., 1983.

Ives, B., Hamilton, S., and Davis, G. B., "A Framework for Research in Computer-Based Management Information Systems," *Management Sciences*, Vol. 26, No. 9, September 1980, pp. 910-934.

Johansen, R. and Bullen, C., "Thinking Ahead: What to Expect from Teleconferencing," *Harvard Business Review*, March/April 1984, pp. 4-10.

Konsynski, B. R. and Nunamaker, J. F., Jr., "PLEXSYS: A System Development System," in Couger, Colter, and Knapp (eds.), *Advanced Systems Development/Feasibility Techniques*, John Wiley & Sons, Inc., 1982.

Ledgard, H., *Software Engineering Concepts*, Addison-Wesley Publishing Co., 1987, pp. 111-127.

Lyytinen, K. J., "Implications of Theories of Language for Information Systems," *MIS Quarterly*, Vol. 9, No. 1, March 1985, pp. 61-74.

Macro, A. and Buxton, J., *The Craft of Software Engineering*, Addison-Wesley Publishing Company, 1987.

Mahmood, M. A., "System Development Methods - A Comparative Investigation," *MIS Quarterly*, September 1987, pp. 293-311.

Mantha, R. W., "Data Flow and Data Structure Modeling for Database Requirements Determination: A Comparative Study," *MIS Quarterly*, December 1987, pp. 531-545.

Martin, J. and McClure, C., *Structured Techniques: The Basis for CASE*, revised edition, Prentice-Hall, 1988.

Naur, P., Randell, B., and Buxton, J. N., *Software Engineering: Concepts and Techniques*, Mason/Charter Publishers, Inc., 1976.

Norman, R. J. and Nunamaker, J. F., Jr., "CASE Productivity Perceptions of Software Engineering Professionals," *Communications of ACM*, Vol. 32, No. 9, September 1989, pp. 1102-1108.

Nunamaker, J. F., Jr. and Chen, M., "Software Productivity: A Framework of Study and Approach to Reusable Components," *The 22nd International Conference on System Sciences*, Hawaii, January 1989.

Orlikowski, W. J., "CASE Tools and the IS Workplace: Findings from Empirical Research," Proceedings of the 1988 ACM SIGCPR Conference, 1988, pp. 88-97.

Parnas, D. L., "Designing Software for Ease of Extension and Contraction," *IEEE Transactions on Software Engineering*, Volume SE-5, Number 2, March 1979, pp. 128-138.

Roadstrum, W. H., *Excellence in Engineering*, John-Wiley & Sons, Inc., 1967.

Scott Morton, M. S., "The State of Art of Research," in F. W. McFarlan (ed.), *The Information Systems Research Challenge*, Harvard Business School Press, 1984, pp. 13-41.

Vick, C. R. and Ramamoorthy, C. V., *Handbook of Software Engineering*, Van Nostrand Reinhold, 1984.

Wegner, P., "Paradigms of Information Engineering," in F. Machlup and and U. Mansfield (eds.), *The Study of Information*, John Wiley & Sons, Inc., 1983, pp. 163-175.